Problem A. WonderTeam

The *Brasileiro League* is the most important event in Brazil. There are n football teams participating in the competitions, each team plays twice (home and away) against each other team. Each team receives three points for a win and one point for a draw. No point is awarded for a loss.

When the games are finished, teams are ranked by numbers from 1 to n according to the total points. The rank of each team t having p points is one plus the number of teams having more than p points. It is possible that more than one team have the same ranks.

In addition to the Champion (the 1st ranked team or teams), the *WonderTeam* is also awarded, if there exists one. The team that has absolutely the highest number of wins (absolutely means no other teams has the same number of wins), absolutely the highest number of goals scored, and absolutely the lowest number of goals conceded, is called the WonderTeam. (WonderTeam should have all these properties.)

Your task is to find out the worst possible rank for the WonderTeam.

Input (Standard Input)

There are multiple test cases in the input. Each test case consists of only one line containing n ($1 \le n \le 50$), the number of teams in league. The input terminates with a line containing 0.

Output (Standard Output)

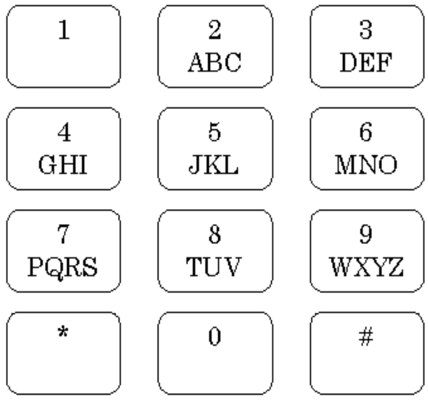
For each test case, write a single line containing the worst possible rank for the WonderTeam.

Sample Input and Output

Standard Input	Standard Output
1 3 0	1 1

Problem B. Encrypted SMS

This year, ACM scientific committee members use emails to discuss about the problems and edit the selected ones. They know that email is not a secure way of communication, especially on such an important topic. So they transfer password-protected compressed file among themselves. In order to send the passwords, they use SMS. To increase the security level, the encrypted passwords are sent by SMS. To do this, a *multi-tap* SMS typing method is used.



The standard 12 key telephone keypad

Multi-tap is currently the most common text input method for mobile phones. With this approach, the user presses each key one or more times to obtain the wanted characters. For example, the key 2 is pressed once to get character A, twice for B, and three times for C.

The encryption algorithm that is used is quite simple: to encrypt the i^{th} character of the password, the key used to obtain that character is tapped i more times. For if the 4th character of password is U, the key 8 is tapped 6 times, getting the character V. Note that to make the problem simple, we have assumed that the keypad does not generate digits.

The scientific committee needs a program to decrypt the received passwords. They are too busy to write this program and have asked you to help! Write a program to get a correct encrypted text and print the original password.

Input (Standard Input)

The input consists of multiple test cases. Each test case contains a non-empty string of length at most 100, consisting of small or capital English letters. The last line of the input contains a single #.

Output (Standard Output)

For each test case, write the decrypted password in a separate line. Note that passwords are case-sensitive.

Sample Input and Output

Standard Input	Standard Output
BACE GgaudQNS #	ABCD IhateSMS

Problem C. Hopeless Coach

One of the Premier League (Persian Gulf Cup) teams had very bad results this year. The board is under pressure to fire the coach, but the coach is considered hero by some fans and it is not easy to fire him. The board decides to give him a last chance; they talked to media that they can only support the coach if the team gets at least 11 points in the next 5 matches. The coach wants to know the probability of passing their condition and ask you to help him. You can assume that the probability of having a win/draw/loss in a future match can be determined from the results of the matches the team currently has played. For example, if the team has already played 10 matches and has won three of them, the probability of having a win in any of the next five matches is 30%. The same rule applies to draws or losses.

You also know the team results (a win earns 3 points and a draw earns 1). There are 18 teams in the league and each team play against each of the other teams twice.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains two numbers *N* and *P*. *N* is the number of matches and *P* is the points that are required in the next *N* games. This is followed by three numbers W, D and L (the number of wins, draws and losses in the previous games). The last line of the input contains two zero numbers.

Output (Standard Output)

For each test case, you should print the percentage probability of getting at least *P* points in the next *N* matches with exactly one digit after decimal point.

Sample Input and Output

Standard Input	Standard Output
5 11 3 5 4 2 3 5 0 5 3 5 5 5 4 1 1 1 1 1 0 0	4.3 75.0 42.8 66.7

Problem D. Yungom

After getting her Ph.D in Cooking with her research paper on "How to Prepare a Pizza", and another Ph.D in Medicine for finding cures for H.I.V and Alzheimers, Dae Jang Guem (Called Yungom in Persian) decided to solve yet another open problem in Information Theory that even Shanon (the father of Information Theory) failed to solve. She is going to construct a language of n words with d given characters $c_1, c_2, \ldots, c_d$. This language should be prefix free which means that there is no pair of words like (s, t) in which the word s is a prefix of t. Each character c_i has a usage cost of w_i. The cost of a word s with the length l is the sum of the costs of its l characters. For example, if $c_1=a$; $c_2=b$; $w_1=1$ and $w_2=10$, the cost of word "*aba*" is 1+10+1=12. Similarly, the cost of a language with n words is equal to the sum of the costs of its n words. For example, the cost of language "*ab*"; "*bbb*"; "*baaa*" is 11+30+13=54. Like her previous jobs, Yungom is going to do this task perfectly which means that she wants to find the minimum cost, prefix free language with n words.

Input (Standard Input)

There are multiple test cases in the input. Each test case starts with a line containing two integers n ($1 \le n \le 200$) and d ($1 \le d \le 200$). The next line contains nonnegative integers $w_1, w_2, \ldots, w_d$. The input is terminated by a line containing two zero numbers.

Output (Standard Output)

For each test case, you should print the minimum cost of a prefix free language with n words and d characters.

Sample Input and Output

Standard Input	Standard Output
3 4 1 10 100 1000 0 0	23

Problem E. New Island

A new island has been discovered. A team of architects has worked hard and proposed a road plan to connect important parts of this new island. Due to lack of fund, we are to modify the design to come up with an affordable one.

In the proposed plan, each road has a unique id between 1 and E (the number of roads) and a cost that is unbelievably equal to 2^{id}. So, the costs are distinct powers of two. We want to eliminate some of the roads from the plan to get the minimum overall cost while all places are still connected. But, we should not eliminate as many roads as we want. The constraint is that in the new road plan the distance between any two places cannot become more than twice as their distance in the original plan. The distance between two places is the minimum number of roads connecting them. The original road plan is given to you in form of a graph and you are asked to find the most economic road elimination according to the constraint.

Input (Standard Input)

There are multiple test cases in the input. Each test case is started with a line containing two integers N ($1 \leq N \leq 200$) and E, the number of vertices (places) and edges (roads) respectively. The specification of the roads comes on the next E lines. The i^{th} line contains two numbers v_i and u_i which means that the road with id i is between places v_i and u_i. The input is terminated by a line containing two zero numbers.

Output (Standard Output)

For each test case, write the number of eliminated roads followed by the increasing list of their ids on a single line.

Sample Input and Output

Standard Input	Standard Output
4 5 1 2 3 1 4 1 4 2 3 4 3 3 1 2 2 3 3 1 0 0	2 4 5 1 3

Problem F. Sub-dictionary

In this problem, by the word "dictionary" we mean a list of alphabetically ordered words and their associated explanations in the same language. A dictionary must contain the definition for any word that appears in the explanation of another word. So you see, if a dictionary defines N words, it has exactly N distinct words in it. Also, we know that in a dictionary no word appears in the definition of itself.

A sub-dictionary is a collection of dictionary's words and their definitions such that it can be published as an independent dictionary, obviously satisfying the mentioned condition. As a project of Computational Linguistics course, we are assigned to create a Lexical Knowledge Base which is the knowledge expressed by words. For this task we should create our knowledge foundation based on a dictionary.

It's really hard for the computer to study words automatically. So, we decided to manually teach it some common words. We start from an appropriate sub-dictionary. By understanding its words, a computer could extend its knowledge to the whole dictionary word by word. For instance, a word "xyz" could be added to the computer's understanding if computer knows the meaning of every words used in xyz's definition. You are asked to write a program that can find the smallest extendable sub-dictionary for a specific dictionary.

Input (Standard Input)

The input consists of multiple test cases. The first line of each test case is n ($1 \le n \le 100$), the number of dictionary's words. Each of the next n lines contains a word and its definition (that has at most 30 words). Words are separated by blanks and are made up of small English letters less than 25 characters.

Output (Standard Output)

For each test case, on the first line print the number of sub-dictionary's words and on the second line write the alphabetically sorted list of words (separated by blanks).

Sample Input and Output

<table>
<tr><th>Standard Input</th><th>Standard Output</th></tr>
<tr><td><pre>5
aue oizer piqoi oizer
doy oizer hweqlo hweqlo
hweqlo piqoi aue
oizer piqoi
piqoi aue aue
0</pre></td><td><pre>3
aue oizer piqoi</pre></td></tr>
</table>

Problem G. Sharif Super Computer

SSC is a super computer designed in Sharif University having 2 "master" and n "slave" processors. It can run softwares in parallel: One of the master processors loads the software on the slave processors such that the memory and CPU usage among them are balanced, while the other master is used for monitoring the system.

Because of the dependencies between different parts of the software, many messages should be exchanged between processors. A very fast network is needed to minimize the message passing overhead. To optimize the network, a *clique* structure will be constructed in which there is a direct communication cable between each pair of processors.

There are two different cables: blue cables which can transmit up to 100 Megabits per second and red cables which can transmit up to 1 Gigabits per second. Each pair of slave processors will be connected by one blue cable. Due to the higher communication volume on master processors, the two masters are connected by one red cable and also each master is connected to each slave by another red cable.

SSC is thus made of $n + 2$ motherboards, each containing exactly one processor, the needed memory, and also $n + 1$ similar network sockets installed as a horizontal array. The motherboards are put in a vertical rack box, each in one horizontal shelf. So, each motherboard is uniquely identified by its height in the rack.

The cooling system has forced us to put the two master motherboards in the lowest and highest shelves of the rack. We assume that the master in the bottom has height 0, and the heights of the other motherboards are integers higher than 0. You, as a computer engineer, are asked to do the final assembly of SSC. You are given the empty rack box, the ready motherboards, and your job is to determine whether you can put the boards in the rack that satisfy the constraints and cable lengths.

There are exactly $2n+1$ red cables available with the given sizes. However the blue cables are available in m different sizes, and we have unlimited number of cables in each size. You are so careful to keep the cabling between processors tidy and tight, so you want to install the motherboards in the heights such that the size of cable used between each pair of motherboards is exactly equal to the difference between the heights of two boards.

Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains two numbers n ($1 \le n \le 100$) and m ($1 \le m \le 1000$). The second line contains $2n+1$ integers, which are the sizes of Gigabit Ethernet cables. The third line contains m integers which are the sizes of Megabit Ethernet cable groups. The last line of the input contains two zero numbers.

Output (Standard Output)

For each data set you should write $n+1$ integers as the heights of the motherboards in SSC rack box. The first number represents the height of the top master processor, and the remaining n integers are the positions of the slaves in an increasing order. In the case of having multiple solutions write the one with the minimum alphabetical order. If there is no solution write "Impossible".

Sample Input and Output

Standard Input	Standard Output
3 3 3 7 7 10 10 14 17 3 4 7 3 3 3 7 7 10 10 14 17 3 4 8 0 0	17 3 7 10 Impossible

Problem H. Circle Artwork

Circle is an ancient and universal symbol of unity, wholeness, infinity, the goddess, and female power. It is referenced frequently in religion and art. In this problem, we act as a modern artist and would like to draw our painting with points and circles, and clearly colors should be used. First, we put some colored points on the canvas. The goal is to draw a circle for each color C_i, such that every colored point inside or on the boundary of that circle has color C_i. Also, each such circle should have at least two points on its boundary. Note that for some colors, it might be impossible to draw such a circle. In this problem, given a set of colored points, your task is to compute the largest number of colors for which there exists a circle conforming to the above conditions.

Input (Standard Input)

There are multiple test cases in the input. For each test case, in the first line there is a positive integer n ($1 \leq n \leq 100$), which is the number of colored points. This is followed by n lines of the form C_i X_i Y_i where C_i is the color of the i^{th} point and X_i Y_i specify its coordinates. Each color string is made up of at most 20 small English letters. Coordinates are integers between -1,000,000 and 1,000,000. The last line of each test case contains a single 0.

Output (Standard Output)

For each test case, write a single line which contains the largest number of colors for which there exists a circle conforming to the above conditions.

Sample Input and Output

Standard Input	Standard Output
4 red 1 1 blue 1 2 blue 3 2 yellow 3 3 0	1

Problem I. Crazy Bits

Olandicans have invented a strange computer; it has only 12-bit registers to store numbers. And the only command that this computer accepts is SWAP. The Swap function is called with 3 parameters *i*, *j*, and *d*. A call of swap(*i, j, d*) swaps the j^{th} bit of the i^{th} register with its neighboring bit in direction *d* (0: up, 1: right, 2: down, 3: left). For example, swap (2, 3, 1) swaps the 3rd and the 4th bits of the 2nd register and Swap(6, 4, 2) swaps the 4th bits of the 6th and the 7th registers. Olandicans know the initial values of the registers and they want to change them to some other numbers. They asked you to help them find the minimum number of swap calls.

Input (Standard Input)

The input consists of multiple test cases. The first line of each test case is n ($1 \le n \le 16$), the number of registers. The next line contains n integers, where the i^{th} number is the initial value of the i^{th} register. The next line contains n integers, where the i^{th} number is the desired value of the i^{th} register. The input is terminated by a line containing a zero.

Output (Standard Output)

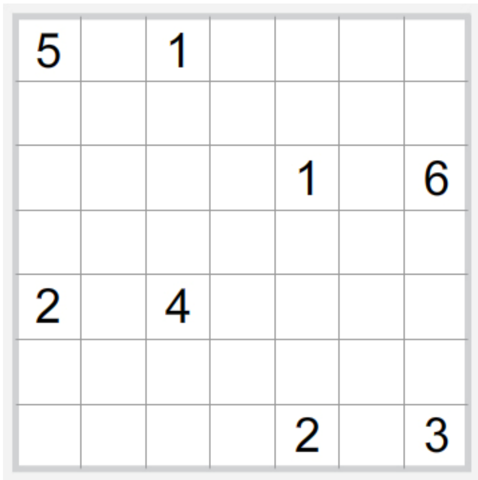
For each test case, you should write a single line containing the minimum number of swaps needed for that test case. If it is not possible, write "Impossible".

Sample Input and Output

Standard Input	Standard Output
2 2 3 6 2 3 1 1 1 2 3 4 4 5 2 6 0 3 2 2 4 0	3 Impossible 2

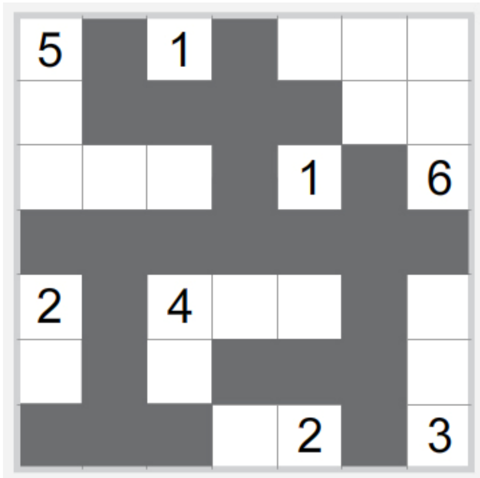
Problem J. Nurikabe

Your goal is to write a solver for Nurikabe, a binary determination puzzle. The puzzle is played on a grid, typically rectangular (with no standard size) containing empty and numbered cells. You must decide for each cell if it is white (land) or black (water), so that it satisfies the following constraints. An Island is a maximal connected region of white cells.



- The water areas must form one connected region. (All the black cells must be connected.)
- Each numbered cell must be part of an island.
- The number of cells in an island is equal to the number it contains.
- Every region (island) of white cells (land) must contain exactly one number.
- Two islands may not be connected.
- 2x2 blocks of black squares are not allowed.

Note that diagonal adjacency doesn't count as connectedness. You can assume there is always a unique solution for each puzzle.



Input (Standard Input)

There are multiple test cases in the input. The first line of each test case contains two numbers *n*, *m* ($3 \le n,m \le 9$) which are the dimensions of the puzzle, followed by *n* lines each one has *m* characters including '.' (indicating an empty cell) and 1-digit numbers. The last line of the input contains two zero numbers.

Output (Standard Output)

The output for each test case should show the solved puzzle. Show black (water) cells with '#'. Write an empty line in the output after each puzzle.

Sample Input and Output

Standard Input	Standard Output
3 4 3... .... .4.. 5 5 2.5.. ....4.3 0 0	3..# #### .4.. 2#5.. .#.## ##.#. .###. ..4#3